

## **Remarks**

The above amendments and these remarks are in reply to the Office Action mailed February 7, 2008, the Advisory Office Action mailed April 29, 2008, the Examiner Interview of May 12, 2008, and the Examiner Interview Summary of May 22, 2008.

### **I. Summary of Rejections**

The Office Action rejected claims 11-21.

Claims 11-20 were rejected under 35 U.S.C. 102(b) as being anticipated by Bogle et al. (U.S. Pat. No. 6,353,923 B1).

Claim 21 was rejected under 35 U.S.C. 103(a) as being unpatentable over Bogle et al. (U.S. Pat. No. 6,353,923 B1).

### **II. Statement of Summary of Examiner Interview of May 12, 2008**

Applicants acknowledge with thanks Examiner Vo's assistance in granting an interview on May 12, 2008, during the course of which interview various features of the claimed embodiments were discussed, the substance of which is included herein and is found in the Examiner Interview Summary of May 22, 2008. Applicants' representatives and Examiner Vo discussed a proposed amendment to Claim 11, and Claim 15 as previously filed.

### **III. The Prior Art Reference Does Not Anticipate or Suggest Claims 11-22 Independent Claim 22**

New independent claim 22 was added to the claims. Claim 22 states:

A system, comprising:

a multi-language debugger, wherein the multi-language debugger debugs a source code file which contains compiled and interpreted languages;

a script engine interface, wherein a script engine communicates to the multi-language debugger through the script engine interface;

a script debug controller;

a script context object, wherein the script engine uses the script context object to hold a script context;

a debuggable frame object, wherein each of the compiled and interpreted languages can be edited in the debuggable frame object;

an interface to a runtime messaging environment, wherein the interface is implemented by a runtime messaging environment that controls a running state of the script engine;

a proxy, wherein the proxy is used between the executing code being debugged and the multi-language debugger to consolidate contents of one or more messages; and

a debug commands interface.

Independent Claim 22 includes features that are not disclosed in Bogle. Amongst other features that are not disclosed, Claim 22 requires “a proxy, wherein the proxy is used between the executing code being debugged and the multi-language debugger to consolidate contents of one or more messages.” The Office Action cited Bogle’s marshalling proxy as disclosing these features in the discussion of Claim 15. Bogle’s marshalling proxy, however, is not used to consolidate the contents of one or more messages. Bogle’s marshalling proxy therefore does not teach or suggest Claim 22’s requirement that the proxy is used between the executing code being debugged and the multi-language debugger to consolidate contents of one or more messages.

35 U.S.C. §102 Rejection to Independent Claim 15

Claim 15 was amended to become an independent claim. Claim 15, as amended, states:

A system, comprising:

a multi-language debugger, wherein the multi-language debugger debugs a source code file which contains compiled and interpreted languages;

a script engine interface, wherein a script engine communicates to the multi-language debugger through the script engine interface;

a script debug controller;

a debuggable frame object, wherein each of the compiled and interpreted languages can be edited in the debuggable frame object;

an interface to a runtime messaging environment, wherein the interface is implemented by a runtime messaging environment that controls a running state of the script engine;

a debug commands interface; and

The system of claim 11, further comprising:

a proxy, wherein the proxy is used between the executing code being debugged and the multi-language debugger to consolidate contents of one or more messages.

Independent Claim 15 includes features that are not disclosed in Bogle. Amongst other features that are not disclosed, Claim 15 requires “a proxy, wherein the proxy is used between the executing code being debugged and the multi-language debugger to consolidate contents of one or more messages.” The Office Action cited Bogle’s marshalling proxy as disclosing these features in the discussion of Claim 15. Bogle’s marshalling proxy, however, is not used to consolidate the

contents of one or more messages. Bogle's marshalling proxy therefore does not teach or suggest Claim 15's requirement that the proxy is used between the executing code being debugged and the multi-language debugger to consolidate contents of one or more messages.

35 U.S.C. §102 Rejection to Independent Claim 11

Independent Claim 11 (as amended) states:

A system for debugging in more than one programming language, comprising:

a multi-language debugger, ~~with the capability to debug~~ wherein the multi-language debugger debugs a source code file which contains ~~multiple nested languages, wherein the multi-language debugger interprets multiple languages that are nested in a single source file, and wherein the multiple nested languages can include both~~ compiled and interpreted languages;

a script engine interface, wherein a script engine communicates to the multi-language debugger through the script engine interface;

a script debug controller, wherein the script debug controller registers itself upon start-up; ~~wherein the multi-language debugger uses a standardized interface for a script engine, wherein all communications with the script engine will be through calls to the script debug controller;~~

a script context object, wherein the script engine can use the script context object to hold a script context;

a debuggable frame object, ~~wherein the script engine uses a debuggable frame object to retrieve script context for a supported language, wherein each of the multiple nested languages is displayed in a debuggable frame object, and wherein each of the multiple nested~~ compiled and interpreted languages ~~can be~~ are edited in the debuggable frame object;

an interface to a runtime messaging environment, wherein the interface is implemented by a

runtime messaging environment that controls a running state of the script engine; and  
a debug commands interface.

Claim 11 includes features that are not shown or made obvious by the cited reference. Claim 11 requires a multi-language debugger, wherein the multi-language debugger debugs a source code file which contains compiled and interpreted languages. The Office Action erroneously argued that Bogle disclosed this feature, even though Bogle's debugger cannot debug a source code file that contains compiled and interpreted languages. The Office Action incorrectly asserted that there was support for its erroneous argument in *Gardner v. TEC Systems, Inc.*, 725 F.2d 1338, 220 USPQ 777 (Fed. Cir. 1984). In *Gardner*, the only difference between the prior art air bar and the claimed air bar was the size of the claimed air bar, and the air bar did not perform or operate any differently from the prior art air bar. *Gardner* does not support the Office Action's position, because there clearly is a difference between "a multi-language debugger, wherein the multi-language debugger debugs a source code file which contains compiled and interpreted languages," and Bogle's debugger, which cannot debug a source code file which contains compiled and interpreted languages. This is not similar to a case where a first air bar has a different size from a second air bar, but otherwise performs and operates identically.

Bogle's invention is focused on debugging an application that includes multiple program components from many different programming language sources, i.e. many different source code files, not a source code file as in claim 11 (see Bogle, col. 2, line 1: "many different programming language sources").

The Office Action cited Bogle's FIG. 4 and the statement in col. 4, lines 10 – 19, for disclosing this portion of claim 11. FIG. 4 shows multiple host processes and the debug managers,

FIG. 4 does not indicate the capability to debug multiple languages in a source code file. Bogle's col. 4, lines 10 – 19, describes a method for debugging a virtual application that includes multiple compiled and interpreted programming language statements. However, col. 4 says nothing regarding an ability to debug compiled and interpreted languages in a source code file. There is a distinction between debugging a virtual application that contains program code in several source code files versus debugging a source code file that contains compiled and interpreted languages.

Furthermore, Bogle's col. 11, lines 25-37 describes Bogle's Machine Debug Manager in greater detail:

“Virtual applications are collections of related documents and code in a single debuggable entity such that separate application components in a continuous line of code can share a common process and/or thread. A virtual application is the aggregate of multiple applications in multiple programming languages. One key role of the machine debug manager is to act as a program language registry that provides a mapping between a given application in the virtual application aggregate and the active debugger IDE 410 that is controlling the virtual application during the debug process. The MDM 411 eliminates the traditional debugging model where the debugger for a given programming language only has knowledge of a specific source and object code mapping.”

This portion of Bogle teaches away from Applicants' claimed invention. Bogle teaches that his application is not focused on a specific source, i.e. not focused on a specific source code file, and instead Bogle's application is focused on the virtual application as a whole, as collections of related documents and source code files were treated as a single application entity in the aggregate.

Applicants respectfully submit that the embodiment as defined in independent claim 11 is neither anticipated by nor obvious in view of Bogle.

Rejections to Claims 12-14 and 16-21

For at least the reasons discussed above, dependent claims 12-14 and 16-21 are also patentable. Dependent claims 12-14 and 16-21 add their own limitations which render them patentable in their own right.

IV. Conclusion

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and a Notice of Allowance is requested. The Examiner is respectfully requested to telephone the undersigned before an advisory action is issued in order to avoid any unnecessary filing of an appeal.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: June 9, 2008

By: /Thomas K. Plunkett/  
Thomas K. Plunkett  
Reg. No. 57,253

Customer No. 23910  
FLIESLER MEYER LLP  
650 California Street, 14<sup>th</sup> Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800